

Problem A. Alien invaders

Input file: *standard input*
Output file: *standard output*
Time limit: 7.8 seconds
Memory limit: 512 mebibytes

Вы атакованы инопланетянами!

Инопланетяне атакуют в соответствии со следующими правилами: всего нападает n инопланетян; i -й из них появляется в момент времени a_i на расстоянии d_i от вас. Он должен быть уничтожен не позднее времени b_i , иначе он пустит в ход своё оружие и Вы будете уничтожены.

Вы вооружены дезинтегратором, для которого можно установить радиус действия R перед каждым выстрелом. При выстреле дезинтегратор уничтожает всех инопланетян, находящихся на расстоянии R или меньше от Вас, при этом расходуется R единиц заряда.

Какое минимальное количество единиц заряда необходимо потратить для того, чтобы уничтожить всех атакующих инопланетян?

Input

Первая строка входа содержит количество тестовых примеров T .

Каждый тестовый пример начинается строкой, содержащей количество инопланетян n ($1 \leq n \leq 300$). i -я из последующих n строк содержит три целых числа a_i, b_i, d_i , ($1 \leq a_i < b_i \leq 10\,000$; $1 \leq d_i \leq 10\,000$) — время появления i -го инопланетянина, время, в течение которого он не будет стрелять и расстояние от Вас, соответственно.

Output

Для каждого тестового примера выведите одну строку, содержащую наименьшее количество единиц заряда, требуемое для того, чтобы уничтожить всех атакующих инопланетян.

Example

standard input	standard output
1	7
3	
1 4 4	
4 7 5	
3 4 7	

Problem D. Dictionary

Input file: *standard input*
Output file: *standard output*
Time limit: 3.1 seconds
Memory limit: 512 mebibytes

По мнению отдельных гуманитариев, словарь типичного программиста состоит из трёх слов, в каждом из которых программист делает ошибки.

Чтобы помочь программистам правильно писать слова, Ассоциация Озабоченных Гуманитариев выпустила «Словарь трёх слов для программистов».

Вы получили этот словарь в качестве гуманитарной помощи, но почти сразу пролили на него свой кофе, так что некоторые символы стали нечитаемыми. Как и в каждом словаре, слова были попарно различны и заданы в лексикографическом порядке.

По оставшимся символам выясните, сколько различных вариантов восстановления словаря существует.

Input

Первая строка содержит одно целое число T — количество тестовых примеров.

Каждый тестовый пример состоит из трёх строк; каждая из этих строк содержит одно непустое слово. Слова состоят из строчных латинских букв и заданы в лексикографическом порядке. Повреждённые буквы заменены знаками вопроса (один знак вопроса соответствует одной букве). Длина каждого слова не превосходит 10^9 .

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — количество различных вариантов замены знака вопроса одной из 26 строчных латинских букв таким образом, чтобы словарь оставался корректным, взятое по модулю $10^9 + 9$.

Example

standard input	standard output
3	42562
?heoret?cal	52
c?mputer	1
?cience	
jagiellonian	
?niversity	
kra?ow	
?	
b	
c	

Problem E. Express As The Sum

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Задано целое число N . Требуется представить его в виде суммы двух или более последовательных целых положительных чисел. Например, $10 = 1 + 2 + 3 + 4$

$$24 = 7 + 8 + 9$$

Если решений несколько, выведите то из них, которое состоит из наименьшего количества слагаемых.

Input

Первая строка входа содержит одно целое число T — количество тестовых примеров.

Каждый тестовый пример состоит из одной строки, содержащей целое число N ($1 \leq N \leq 10^9$).

Output

Для каждого тестового примера выведите одну строку, содержащую требуемое разложение в виде

$$N = a + (a+1) + \dots + b,$$

форматированное в соответствии с примером из условия. Если решения не существует, выведите текст "IMPOSSIBLE".

Example

standard input	standard output
3	IMPOSSIBLE
8	10 = 1 + 2 + 3 + 4
10	24 = 7 + 8 + 9
24	

Problem F. Factory

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Важный узел секретной машины состоит из n нанощестерёнок, занумерованных последовательными целыми числами от 1 до n . Нанощестерёнки отличаются от обычных шестерёнок тем, что размер их зубьев настолько мал, что шестерёнки можно считать обычными кругами. Каждая нанощестерёнка крутится вокруг своего центра.

Никакие две нанощестерёнки не перекрываются (не имеют общих внутренних точек), однако шестерёнки могут касаться друг друга. Если две шестерёнки касаются и одна из них вращается, другая тоже вращается, так как соответствующие зубья зацеплены.

К шестерёнке 1 приложена сила, заставляющая её вращаться со скоростью 1 оборот в минуту по часовой стрелке. Вычислите скорость вращения остальных шестерёнок. Гарантируется, что при вращении механизм не блокируется.

Input

Первая строка входа содержит одно целое число T — количество тестовых примеров.

Каждый тестовый пример начинается со строки, задающей количество нанощестерёнок n ($1 \leq n \leq 1000$). Каждая из последующих n строк содержит три целых числа x , y и r ($-10\,000 \leq x, y \leq 10\,000$; $1 \leq r \leq 10\,000$), где (x, y) — координаты центра шестерёнки и r — её радиус.

Output

Для каждого тестового примера выведите n строк, описывающих вращение шестерёнок. Для каждой шестерёнки, выведите или “ p/q clockwise”, или “ p/q counterclockwise”, где несократимая дробь p/q задаёт количество оборотов нанощестерёнки в минуту. Если $q = 1$, выведите p как целое число. Если шестерёнка не вращается, выведите “not moving”.

Example

standard input	standard output
1	1 clockwise
5	3/2 counterclockwise
0 0 6	2 counterclockwise
6 8 4	3/2 clockwise
-9 0 3	not moving
6 16 4	
0 -11 4	

Problem H. How to Deal With Imp

Input file: *standard input*
Output file: *standard output*
Time limit: 6 seconds
Memory limit: 512 mebibytes

Вы находитесь в магическом магазине. В нём представлено n изделий, каждое из которых спрятано в магическую коробку. i -я коробка стоит c_i монет и содержит объект стоимостью v_i монет. Значения c_i и v_i известны вам заранее.

При этом вы не можете иметь при себе более одного магического предмета. Вы собрались купить самый ценный из предметов... но неожиданно в магазине появился имп. Имп всегда делают какие-нибудь магические пакости. Вот и сейчас имп может применить заклинание, которое разрушает содержимое любой магической коробки. При этом имп применяет заклинание уже тогда, когда вы заплатили за коробку, так что вам придётся покупать следующую коробку... потом ещё одну...

У импа хватает маны на k заклинаний. При этом имп имеет право выбора, уничтожить очередной предмет или оставить его вам (после чего вы покидаете магазин). Вы можете уйти из магазина в любой момент, ничего не купив, или сразу же после того, как купили объект и имп по какой-то причине его не уничтожил.

Требуется максимизировать доход (то есть разницу между стоимостью полученного объекта и понесёнными затратами), если имп стремится её минимизировать и действует оптимально.

Input

Первая строка входа содержит целое число T — количество тестовых примеров.

Каждый тестовый пример начинается строкой, задающей количество объектов n ($1 \leq n \leq 150\,000$) и максимальное число заклинаний у импа k ($0 \leq k \leq 9$). Каждая из последующих n строк содержит описание объекта — реальную стоимость v_i и цену в магазине c_i ($0 \leq v_i, c_i \leq 10^6$).

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — максимальный доход, который Вы сможете получить.

Example

standard input	standard output
1	7
3 1	
10 5	
8 1	
20 12	

Problem J. Joining The Bricks Together

Input file: *standard input*
Output file: *standard output*
Time limit: 15.6 seconds
Memory limit: 512 mebibytes

Задана последовательность белых (W) и чёрных (B) кирпичей. Требуется собрать их в некоторое количество непустых непрерывных блоков, каждый из которых имеет одно и то же отношение белых и чёрных кирпичей.

При этом требуется максимизировать число блоков. Например, рассмотрим следующие примеры

BWWWBB = BW + WWBB (отношение 1:1),

WWWBBBWWWWWWWB = WWB + BBWWWWW + WWB (отношение 3:1).

В обоих случаях разбиения оптимальны с точки зрения количества блоков.

Input

Первая строка входа содержит одно целое число T — количество тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей целое число n ($1 \leq n \leq 10^5$) — количество одноцветных участков. Каждая из последующих n строк содержит целое число k ($1 \leq k \leq 10^9$) и один из символов “W” или “B”, обозначающих, что k кирпичей данного цвета следуют в исходной последовательности подряд. Гарантируется, что общая длина последовательности не превосходит 10^9 .

Output

Для каждого тестового примера выведите одно целое число — наибольшее количество блоков.

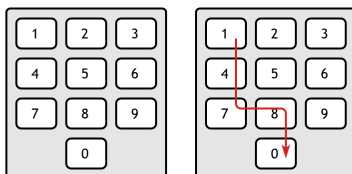
Example

standard input	standard output
3	2
3	3
1 B	5
3 W	
2 B	
4	
3 W	
3 B	
9 W	
1 B	
2	
2 W	
3 W	

Problem K. Keyboard Troubles

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Вы только что проснулись и хотите включить микроволновку на k секунд. Вы должны набрать число k на клавиатуре, имеющей следующий вид:



Так как по сути, вы не совсем проснулись, вы можете набрать число только в случае, если ваши руки будут двигаться вниз или вправо. Вы не можете двигаться влево или вверх, но вы можете нажимать одну и ту же кнопку многократно.

Выведите число, которое Вы можете набрать и которое наиболее близко к k . Если решений несколько, выведите любое из двух.

Input

Первая строка содержит одно целое число T — количество тестовых примеров.

Каждый тестовый пример содержит одно целое число k ($1 \leq k \leq 200$) — требуемое количество минут.

Output

Для каждого тестового примера в отдельной строке выведите ближайшее к k число, которое может быть введено в соответствии с условием задачи.

Examples

standard input	standard output
3	180
180	80
83	133
132	

Problem L. Factory 2

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 256 mebibytes

На фабрике установлены N станков. Каждый станок производит идентичные детали. Для каждого станка известно время в минутах, за которое он производит одну деталь. Станки работают независимо друг от друга. Найдите наименьшее время в минутах, за которое можно сделать K деталей; при этом можно использовать как один станок, так и несколько станков сразу.

Input

Первая строка ввода содержит одно целое число T — количество тестовых примеров ($1 \leq T \leq 700$). Первая строка каждого тестового примера содержит два целых числа N — количество станков ($1 \leq N \leq 10^5$) и K — количество деталей, которые нужно сделать ($1 \leq K \leq 10^9$). Вторая строка содержит N целых положительных чисел t_i — время в минутах, затрачиваемое на производство одной детали на i 'м станке. Все t_i не превосходят 10^9 , сумма всех N во входном файле не превосходит $2 \cdot 10^9$.

Output

Для каждого тестового примера выведите в отдельной строке одно целое число — наименьшее время в минутах, требуемое для производства K деталей.

Example

standard input	standard output
1	2
2 1	
2 2	

Problem M. C--

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Язык C-- был разработан специально для студентов, которые не справились со стандартным курсом по компиляторам.

Программа на C-- представляет собой список команд. Существуют следующие команды:

-
- { — начало блока;
- } — конец блока;
- `int var` — описание переменной `var`;
- `var1=var2` — присвоить значение переменной `var2` переменной `var1`;
- `var=constant` — присвоить переменной `var` значение `constant`;
- `print var` — вывести значение `var`.

Все имена переменных состоят из одной строчной латинской буквы. Константы представляют собой целые числа в диапазоне от 0 до 10^9 . Пробелы допустимы только после операторов “`int`” и “`print`”.

Каждая переменная в блоке описывается не более одного раза, однако переменные с одним и тем же именем могут быть описаны в различных блоках. В этом случае имя задаёт переменную, описанную в ближайшем открытом блоке. Переменные могут быть описаны в середине блока.

Гарантируется, что программа корректна, то есть баланс и вложенность скобок соблюдены, обращение к переменным происходит только после их описания, перед чтением переменные всегда инициализированы и так далее. Программа всегда содержит как минимум один оператор “`print`”.

Напишите программу, исполняющую код на языке C--.

Input

Первая строка входа содержит целое число N ($3 \leq N \leq 1000$). Последующие N строк содержат команды языка C--, по одному на строку.

Output

Для каждой команды “`print`” в отдельной строке выведите печатаемое этой командой значение.

Example

standard input	standard output
10	60
int a	50
int x	
{	
a=50	
int a	
a=60	
x=a	
print x	
}	
print a	

Problem N. Allo

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Задан телефонный номер, составленный из цифр от 0 до 9. Найдите все цифры, которые не используются в этом номере.

Input

Первая строка входа содержит одно целое число T — количество тестовых примеров ($1 \leq T \leq 150$). Каждая из последующих T строк задаёт один тестовый пример — непустую строку, состоящую из цифр от 0 до 9. Длина заданной строки не превосходит 30.

Output

Для каждого тестового примера выведите в возрастающем порядке все числа, которые не используются в соответствующей строке. Если все цифры используются, то выведите текст “allo”.

Examples

standard input	standard output
2	013579
2468	allo
0123456789	

Problem O. Game

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Алиса и Боб играют в следующую игру. Первоначально задано число 1. Игроки делают ходы по очереди, в свою очередь хода каждый игрок должен умножить текущее число на одно из целых чисел от 2 до 9 включительно. Выигрывает тот, кто первый получает число, не меньшее заданного N . Алиса всегда начинает первой.

Для заданного N найдите, кто выиграет при оптимальной игре обеих сторон.

Input

Первая строка входа содержит одно целое число T — количество тестовых примеров ($1 \leq T \leq 2500$). Каждая из последующих T строк содержит одно целое число N ($2 \leq N \leq 10^9$).

Output

Для каждого тестового примера выведите в отдельной строке 1 в случае победы Алисы и 2 в случае победы Боба.

Example

standard input	standard output
4	1
9	2
10	2
1149729	1
999999999	